



US 20040054691A1

(19) **United States**(12) **Patent Application Publication**
Sharma et al.

(10) Pub. No.: US 2004/0054691 A1

(43) Pub. Date: Mar. 18, 2004

(54) **CONTEXTUAL SEARCH INTERFACE FOR
BUSINESS DIRECTORY SERVICES****Related U.S. Application Data**

(60) Provisional application No. 60/387,295, filed on Jun. 7, 2002.

(75) Inventors: Jayant Sharma, Nashua, NH (US);
Ping Wang, Stanford, CA (US);
Lijuan Qian, Lancaster, MA (US); Ji
Yang, Nashua, NH (US)**Publication Classification**(51) Int. Cl.⁷ G06F 17/00; G06F 15/16

(52) U.S. Cl. 707/104.1; 709/203

Correspondence Address:**HAMILTON, BROOK, SMITH & REYNOLDS,
P.C.****530 VIRGINIA ROAD****P.O. BOX 9133****CONCORD, MA 01742-9133 (US)**(57) **ABSTRACT**

Business directory services allow definitions of custom category hierarchies and different search methods, such as, for example, search by location, distance from an addressable location, partial or complete business name, and category. A stepwise refinement search interface provides capability for searching within search results using different search methods or categories. Thus, a user does not have to refine and re-enter search criteria in order to get a progressively selective search. This is particularly useful for mobile phone or hand-held device users who would prefer not to have to re-enter the criteria on each search but rather incrementally refine the search criteria.

(73) Assignee: Oracle International Corporation,
Redwood Shores, CA

(21) Appl. No.: 10/457,830

(22) Filed: Jun. 9, 2003

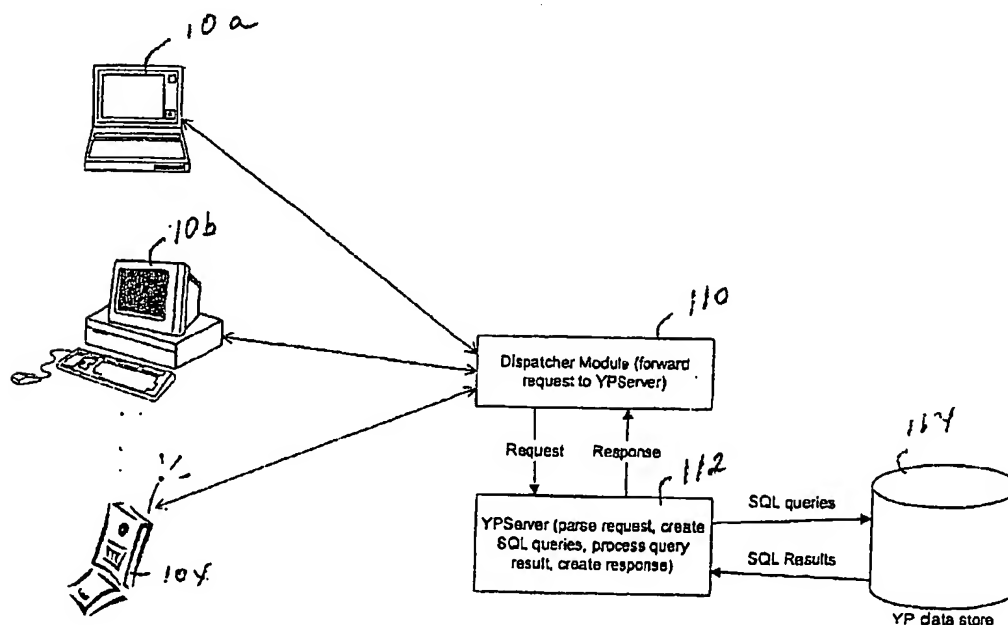


Fig. 1

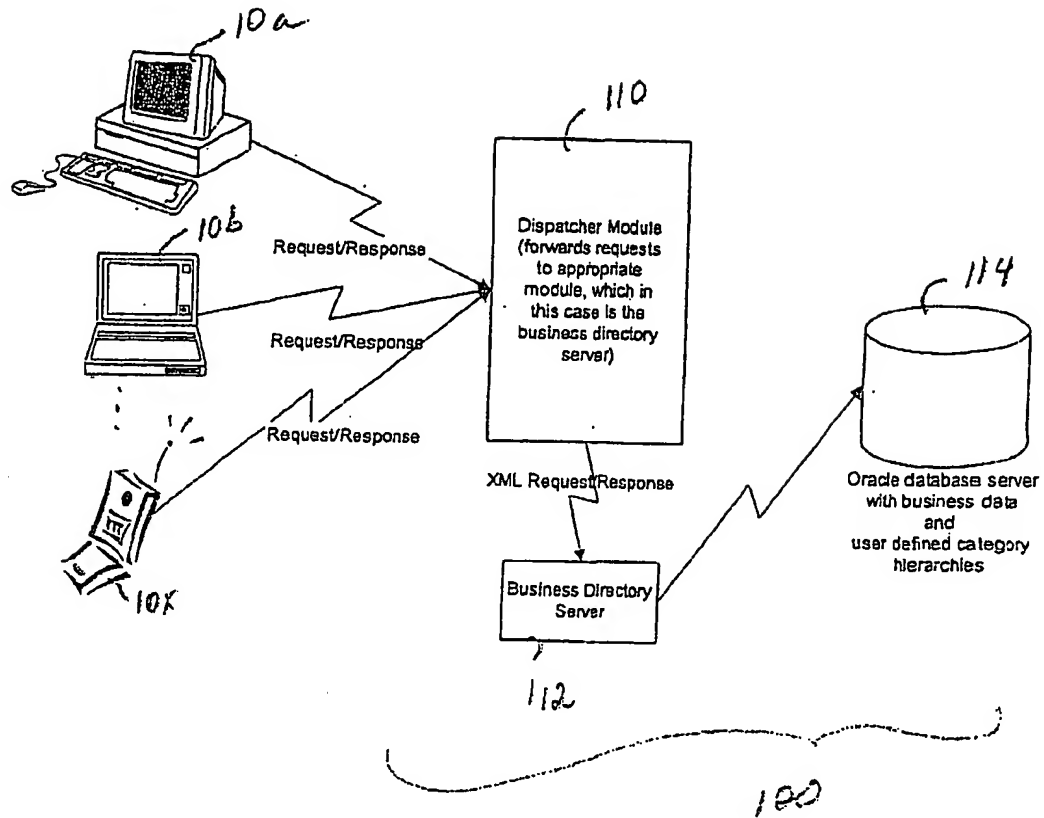
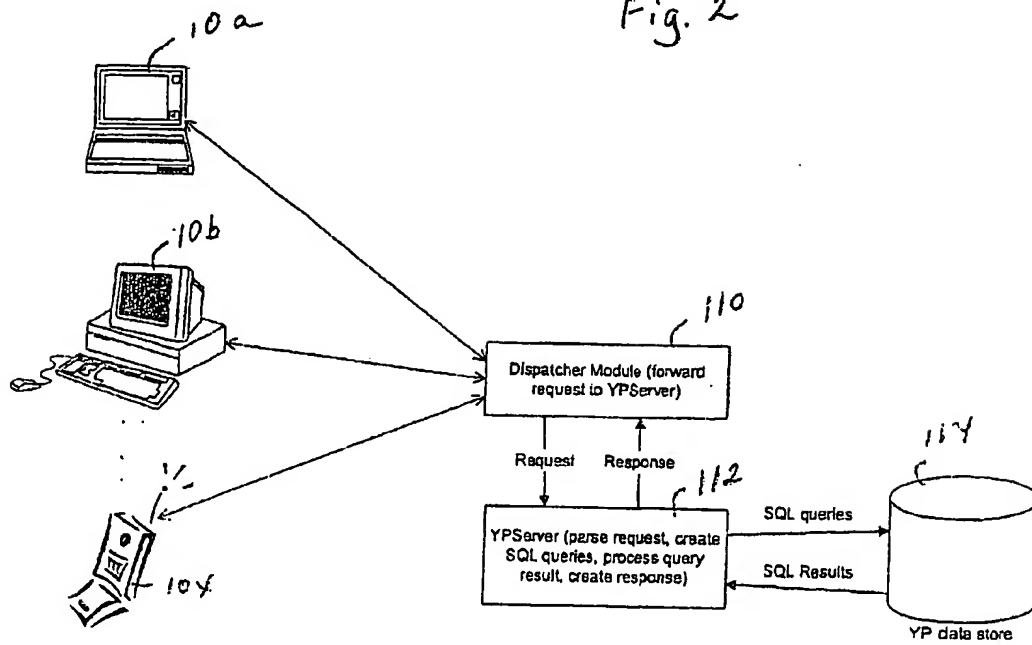


Fig. 2



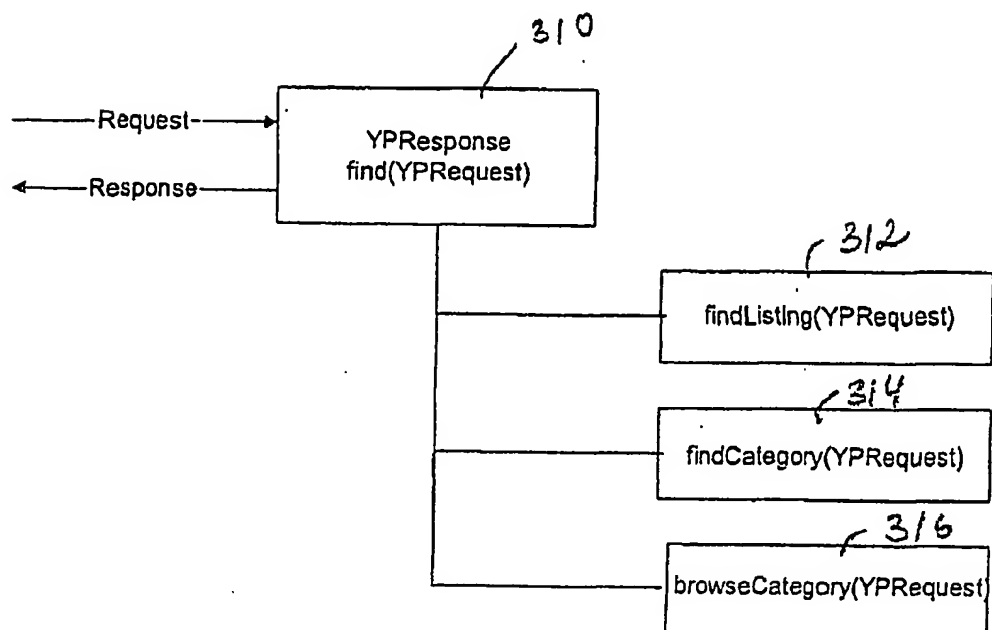


Fig. 3

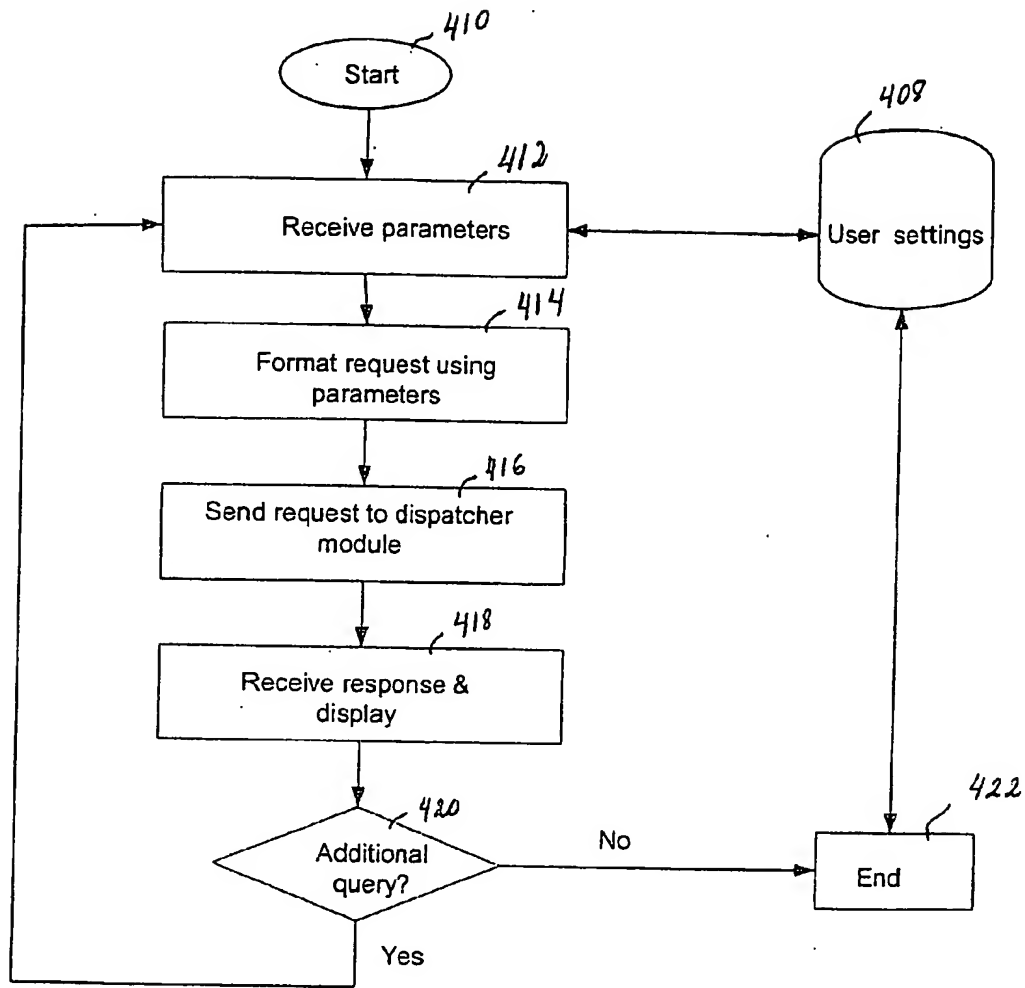


Fig. 4

A sample listing lookup YP request is shown below.

```
510 <yp_request data_provider="infousa" type="listing_lookup">
    512 <listing_lookup_result_type="basic" number_to_return="20">
        514 <category_code_list="12345;23456" />
        516 <listing_name
            518 search_string="BURGER KING"
            search_mode="start_with"
        />
        <search_region>
            <input_location>
                <input_address>
                    <us_form2 postal_code="03062" />
                </input_address>
            </input_location>
        </search_region>
    </listing_lookup>
</yp_request>
```

Fig. 5a

*Fig 5b*Input DTD

```

<!ELEMENT yp_request (listing_lookup | category_lookup |
category_browse)>
<!ATTLIST yp_request
  data_provider CDATA #IMPLIED
  country CDATA #IMPLIED
  type (listing_lookup | category_lookup | category_browse)
#REQUIRED
>
<!ELEMENT listing_lookup (category?, listing_name?,
search_region?)>
<!ATTLIST listing_lookup
  result_type (basic | detail1 | detail2 | detail3) #REQUIRED
  number_to_return CDATA #REQUIRED
>
<!ELEMENT category_lookup EMPTY>
<!ATTLIST category_lookup
  search_string CDATA #REQUIRED
  search_mode (equal | contain | start_with | contain_all |
contain_any | sound_like) #REQUIRED
>
<!ELEMENT category_browse EMPTY>
<!ATTLIST category_browse
  browse_type (root_nodes | all_children | direct_children |
ancestors | siblings) #REQUIRED
  starting_node_id CDATA #IMPLIED
>
<!ELEMENT category EMPTY>
<!ATTLIST category
  code_list CDATA #IMPLIED
  search_string CDATA #IMPLIED
  search_mode (equal | contain | start_with | contain_all |
contain_any | sound_like) #REQUIRED
>
<!ELEMENT listing_name EMPTY>
<!ATTLIST listing_name
  search_string CDATA #REQUIRED
  search_mode (equal | contain | start_with | contain_all |
contain_any | sound_like) #REQUIRED
>
<!ELEMENT search_region (input_location)>
<!ATTLIST search_region
  radius CDATA #IMPLIED
  nearest_n CDATA #IMPLIED
>
<!ELEMENT input_location (point | linestring | input_address)>
<!ELEMENT point EMPTY>
<!ATTLIST point
  srsName CDATA #REQUIRED
>
<!ELEMENT linestring EMPTY>
<!ATTLIST linestring
  srsName CDATA #REQUIRED
>
<!ELEMENT input_address >

```

```
<!--ELEMENT yp_response (category_list | listing_list |
geocode_result)-->
<!--ATTLIST yp_response
  type (category_list | listing_list) #REQUIRED
  error_message CDATA #IMPLIED
-->
<!--ELEMENT geocode_result (location)-->
<!--ELEMENT category_list (category+)-->
<!--ELEMENT listing_list (listing+)-->
<!--ATTLIST listing_list
  number_of_return CDATA #REQUIRED
  result_type (basic | detail1 | detail2) #REQUIRED
-->
<!--ELEMENT category (category+)-->
<!--ATTLIST category
  name CDATA #REQUIRED
  id CDATA #IMPLIED
  category_code CDATA #REQUIRED
  parent_id CDATA #REQUIRED
  level CDATA #IMPLIED
-->
<!--ELEMENT listing (category, location)-->
<!--ATTLIST listing
  name CDATA #REQUIRED
  telephone CDATA #REQUIRED
  fax CDATA #REQUIRED
  category_code CDATA #REQUIRED
  website CDATA #IMPLIED
  stock_symbol CDATA #IMPLIED
-->
<!--ELEMENT location (output_address)-->
<!--ATTLIST location
  longitude CDATA #REQUIRED
  latitude CDATA #REQUIRED
-->
<!--ELEMENT output_address EMPTY-->
<!--ATTLIST output_address
  street CDATA #REQUIRED
  sub_area CDATA #REQUIRED
  builtup_area CDATA #REQUIRED
  order1_area CDATA #REQUIRED
  postal_code CDATA #REQUIRED
  postal_addon_code CDATA #REQUIRED
  house_number CDATA #REQUIRED
  country CDATA #REQUIRED
-->
```

Fig. 5C

COLUMN_NAME	TYPE	PURPOSE
VENDOR	VARCHAR2(20)	Name of the data vendor (e.g. INFOUSA)
COUNTRY_CODE_2	VARCHAR2(2)	2 letter ISO country code of the country to which the data belong to (e.g. US)
CATEGORY_TABLE	VARCHAR2(30)	Name of the category table for this vendor/country (e.g. CATEGORY_INFOUSA)
LISTING_TABLE	VARCHAR2(30)	Name of the listing table for this vendor/country (e.g. LISTING_INFOUSA)

Fig. 6a

COLUMN NAME	TYPE	PURPOSE
ID	NUMBER	Unique identifier
COMPANY_NAME	VARCHAR2(30)	Company name
ADDRESS	VARCHAR2(30)	Street address (e.g. 1 Oracle Drive)
SUB_AREA	VARCHAR2(20)	Name of sub area inside a city in which the company HQ is located.
CITY	VARCHAR2(20)	City in which the company HQ is located
REGION	VARCHAR2(20)	Region in which the company HQ is located. (e.g. this column stores the state names for US listings)
POSTAL_CODE	VARCHAR2(6)	Postal code
POSTAL_ADDON_CODE	VARCHAR2(6)	Postal code extension
PHONE	VARCHAR2(12)	Phone number
FAX_PHONE	VARCHAR2(12)	Fax number
SELECTED_SIC_CODE	VARCHAR2(6)	Assigned SIC code
GBOM	SDO_GEOMETRY	Location of company HQ in Latitude/Longitude
COUNTRY_CODE_2	VARCHAR2(2)	2 letter ISO country code

Fig 6b

COLUMN NAME	TYPE	PURPOSE
ID	NUMBER	Unique identifier
CATEGORY	VARCHAR2(100)	Category Name
SIC	VARCHAR2(16)	Standard Industrial Classification code
PARENT_ID	NUMBER	ID of parent of this category

Fig 6c

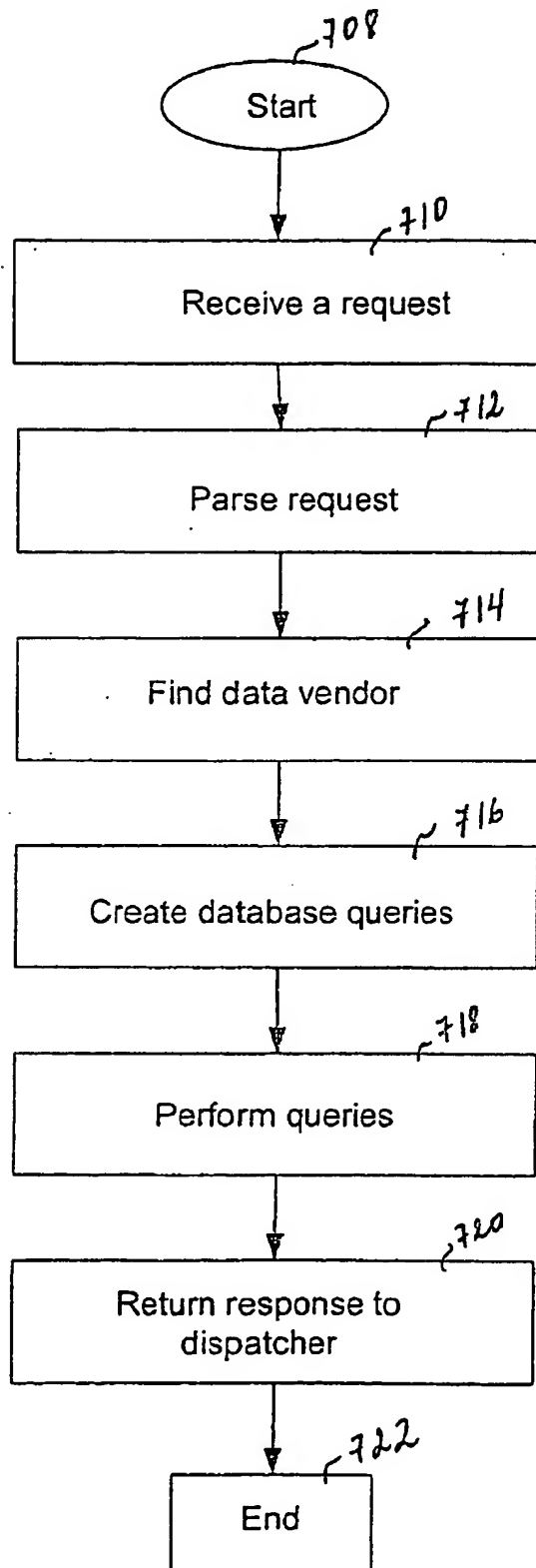


Fig. 7

CONTEXTUAL SEARCH INTERFACE FOR BUSINESS DIRECTORY SERVICES

RELATED APPLICATION(S)

[0001] This application claims the benefit of U.S. Provisional Application No. 60/387,295, filed on Jun. 7, 2002.

[0002] The entire teachings of the above application are incorporated herein by reference.

BACKGROUND

[0003] Business directory services provide users with lists or names of businesses in a specified category or location. A user can request a business of a particular type in a particular city and receive a listing of all available businesses of that type. Such services are sometimes referred to as Yellow Page services. Existing directory services have different interfaces and functionality: some allow searching by partial or exact business name, within a city, or by category, or a combination of the three. A smaller set of service providers allow searching by name and category within a specified distance from an addressable location. An addressable location is a uniquely identified geographical point such as, for example, a street address or a uniquely identified point of interest, or named administrative areas such as cities or postal codes.

[0004] There are several standard nomenclatures for categorization and classification of businesses. Examples of such standards are the Standard Industrial Classification (SIC) codes, the North American Industrial Classification System (NAICS), which replaces the SIC system, and Universal Standard Products and Services Classification created by the Electronic Commerce Code Management Association (ECCMA). In addition, some directory service providers create custom nomenclatures and protocols.

[0005] Business directory services are accessed by users from computers or other electronic access devices, such as, for example, cellular phones enabled for web browsing. If not satisfied with results of the first query, a user can submit another query, supplying original parameters as well as additional restrictions. This narrowing of search, referred to as "drilling down," is typically limited to the original category or subcategory. For example, if the user is looking for a car dealership within a certain distance of a given location, in a process of drilling down, the user may be able to narrow the allowed distance but not add additional markers or specifications for the search.

SUMMARY

[0006] Particular embodiments of the invention can include business directory services that allow definitions of custom category hierarchies and different search methods, such as, for example, search by location, distance from an addressable location, partial or complete business name, and category. A stepwise refinement search interface can provide a capability for searching within search results using different search methods or categories. Thus, a user does not have to refine and re-enter search criteria in order to get a progressively selective search. This is particularly useful for users of wireless devices, such as mobile phone or hand-held devices, when users may prefer not to have to re-enter the criteria on each search, but rather incrementally refine the search criteria.

[0007] Additional functionality can allow customers to browse business categories and drill down a search from a simple cell phone interface, where entering a lot of text at the same time is not practical. A context object can be used to represent search criteria at any given time. The context object can meet the requirements listed above by, for example, allowing users to specify partial search criteria at any given time. With the narrowed result set returned, the user can continue to specify more criteria and search within the results until the final page entry is located.

[0008] Aspects of the invention include methods for providing business directory services to users. A user can enter query parameters into a client. The search parameters can be stored in a context object and passed to the business directory server, which process the requests and returns search results to the client. Once the search results are presented to the user, the user can decide to revise the query by drilling down into particular listings or categories. Additional search parameters can then be added to the context object and can be passed back to the business directory server for processing. Searching can proceed in such iterations until the desired level of result granularity is achieved.

[0009] Search parameters can include type of the search to be performed: whether it is to find a listing, browse listings, or browse by category. Within each type, there may be additional limitations. In addition, user can specify search parameters based on a particular geographic object (for example, an address, or a uniquely defined geographic location).

[0010] If the user is accessing the business directory services from a wireless device, the results can be presented in such a way as to make it easy for the user to browse them and to enter additional search parameters.

[0011] The business directory services system can consist of a dispatcher module, which receives requests from the client, a business directory server module, which processes the requests and sends database queries to a database. The business directory server module also processes database results, converts them to a pre-defined XML schema and sends them back to the client through the dispatcher module.

BRIEF DESCRIPTION OF THE DRAWINGS

[0012] The foregoing and other objects, features and advantages of the contextual search interface for business directory services will be apparent from the following more particular description of particular embodiments of the invention, as illustrated in the accompanying drawings in which like reference characters refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead being placed upon illustrating the principles of the invention.

[0013] FIG. 1 is a schematic diagram of the business directory service architecture according to one embodiment of the invention;

[0014] FIG. 2 is a schematic diagram illustrating processing of a request;

[0015] FIG. 3 is a schematic diagram of the business services server;

[0016] FIG. 4 is a flow chart illustrating operation of the business directory services request client;

[0017] FIGS. 5a-c illustrate XML schema for requests and responses between clients and the business directory server;

[0018] FIGS. 6a-c illustrate database schema for a business directory database;

[0019] FIG. 7 is a flow chart illustrating operation of the business directory server module.

DETAILED DESCRIPTION

[0020] The use of business directory services is greatly improved with additional functionality that allows users to refine and improve search criteria based on the received search results and additional categories. For example, if a user wants to find a Mercedes-Benz dealer around the Boston area, the user can specify "automobile dealers" as a high level category. The business directory service engine might then return a list of several categories, from which the user can select "new car dealers." A list of car brands will be presented next, with the context "automobile/new car dealers" as a business category. Then the user can select "Mercedes-Benz" followed by the location, for example, Boston, Mass., as the location criteria. During each query step, an object containing context, that is, the business category hierarchy and the location hierarchy (or simply location information) is recorded and passed on to the following inquiries.

[0021] Implemented in a particular embodiment of the invention are various search features, such as looking up a business listing by a combination of name, category and location. For example, one may look at a listing of a business names "Pizzico" in the category "restaurants" (or eating and drinking places) and in Nashua, N.H. In addition specific business or home locations can be looked up or browsed by the category hierarchy. Additional search features can be supported such as, for example, searching by names of individuals (instead of businesses) or searching using custom hierarchies.

[0022] Various string match modes and a business or category name look-ups can be supported. The modes are: "exact," "starts with," "contains," and "sounds like." In addition, searches like "nearest neighbor" and "within distance" are supported when a location is specified. That is, the user can run a search for a business within a specified category or a specific business that is within a specified distance from or nearest to a given location. The location can be described as a place name, street address, postal code or geographic objects.

[0023] Data for the business directory service can be provided by multiple providers, each using its own data format. Different data formats can be accommodated as described below in connection with FIG. 6, and multiple data providers' classification schemes or category hierarchies can be supported within one business directory service implementation.

[0024] As used herein, "business directory service" refers not only to services allowing users to search for various businesses, but also to a more generalized search service, allowing users to search for addresses, locations defined by description, businesses or any other geographical object. A particular business directory service is implemented using Standard Industrial Classification (SIC) codes, however, an

alternative embodiment of the invention can use any other standard or custom categorization nomenclature.

[0025] Referring now to FIG. 1, there is shown a schematic representation of a general architecture of the business directory service according to one embodiment of the invention. The business directory service 100 is implemented as a multi-tier system. Clients 10a-x access the system 100 using networked clients using communication protocols such as known in the art. Networked clients 10a-x can be running on personal computers, mobile phones, hand-held devices, or any other devices capable of being connected to a network.

[0026] The system 100 consists of a dispatcher module 110, which receives requests from clients and performs initial processing. The dispatcher module 110 is connected to a business directory server 112. The business directory server 112, in turn, interacts with a database server 114 in order to fulfill clients' requests. The database server 114 can contain business data and user-defined category hierarchies. The data contained in the database server 114 can come from different data providers, and be formatted in different ways, depending on the type of information contained in a particular subset of data.

[0027] The operation of the system 100 is described in further detail in FIG. 2. The clients 10a-x send requests to the dispatcher module 110, which forwards those requests to the business directory server 112. The business directory server 112 parses the request and creates database queries. In response to a database query sent by the business directory server 112, the database server 114 returns appropriate database results. The business directory server 112 then formats the received results and forwards them back to the dispatcher module 110, which, in turn, communicates with the requesting client. In a particular embodiment of the invention, the business directory server 112 communicates with the database server 114 using SQL queries. In an alternative embodiment of the invention, server/database communications can be implemented in any appropriate way, as determined by one skilled in the art. The database server 114 can include additional servers or multiple databases, all referred herein to as a single database.

[0028] The dispatcher module 110 and the business directory server 112 can be separate programs running on different computers, or they can be located on a single computer, with only logical separation in their function. In addition, the database 114 can also be located on the same physical hardware, or, alternatively, be spread out over several physical servers.

[0029] Operation of the business directory server 112 is further described in connection with FIG. 3. The business directory server module 112 consists of two subroutines: an initialization subroutine (not shown) and a find subroutine 310. Initialization subroutine is used to load data vendor information from the database 114 during the initialization time. Loading the database vendor information ensures that the business directory server 112 can issue SQL queries to the right tables by looking up in memory the appropriate vendor information tables.

[0030] After initialization, the business directory server 112 is ready to process the client requests. When the system 100 receives a business directory request, it forwards it to

the business directory server 112. A request is received at the find subroutine 310. In one embodiment of the invention, the business response module is implemented in Java, although a different programming language can be used, as determined by one skilled in the art. The business directory server 112 runs on a web application server (for example, Oracle 9i Internet Application Server). The dispatcher module 110 runs as a separate servlet (for example, as a J2EE component) and receives all the requests. In an alternative embodiment of the invention, both the business directory server 112 and the dispatcher module 110 can be implemented as separated servers running on separate hardware.

[0031] The business directory find subroutine 310 determines whether a particular request is for looking up a listing by business name, or looking by category name, or by browsing by category hierarchy. It then passes the request on to the proper function, such as a find listing function 312, a find category function 314, or a browse category function 316, respectively. In an alternative embodiment of the invention, additional find or browsing capabilities may be provided, such that the user can, for example, browse individual listings or be able to select from different data vendors. Once the results are returned from the database 114, the find subroutine 310 sends results to the dispatcher module 110 to be returned to the client 10.

[0032] The clients 10a-x can be implemented using software appropriate for the particular client type. For example, client software for the client 10x, a cellular phone, can be implemented to have minimum display requirements and be able to fit as much information on a small display, as possible. Regardless of the type of client used, client software on all clients can take similar steps in sending and receiving requests. Operation of client software modules is generally illustrated in FIG. 4.

[0033] After initializing the business directory request module in the step 410, the user can enter request parameters in the step 412. These request parameters can include, for example, business location, business category, or any other search parameters. In an alternative embodiment of the invention, initial request parameters can be retrieved from a user settings store 408, such that the user does not have to define initial parameters. This can be used for, for example, initiating a query from a particular geographic location or limit to this location. In addition, the user settings 408 can contain the parameters of the last search performed by this client.

[0034] In another embodiment of the invention, some of the initial search parameters can be supplied from external modules—for example, in a client equipped with GPS capabilities, the initial search parameters can automatically include location within a certain distance from the present location of the client. In yet another embodiment of the invention, similar location capabilities of the cellular phones may be used to supply geographic restrictions for initial search parameters. Obtaining geocode parameters is described, for example, in U.S. patent application Ser. No. 10/165,811, filed on Jun. 7, 2002, the entire teachings of which are incorporated herein by reference.

[0035] After the search parameters are received in step 412, the client formats the proper request using those parameters in step 414. The request is sent to the dispatcher module 110 in step 416. The client then awaits receipt of the

response from the dispatcher module 112. Received results are displayed in step 418. Based on the results received from the initial query, the user may decide to drill down in to the categories or to add additional search parameters. If the decision to drill down is made in step 420, the module returns to the step 412 where new parameters now include those that the user chooses to enter in addition to the original query. The client module can proceed in such iterations until the user receives the desired results.

[0036] A context object is used within the client to store information about the current query—when the user enters the search parameters for drilling down, they are added to the context object. In such a way, additional information is maintained from one query to another, and the user does not need to re-enter all the original parameters. Information from the context object can then be sent to the business directory service 100 in order to obtain appropriate results.

[0037] In a particular embodiment of the invention, all the query parameters get sent from the clients 10a-x to system 100 in all queries. In an alternative embodiment of the invention, some customization or caching can be done within system 100. For example, business directory server 112 can store information about previous queries from particular clients. In this case, the client 10 would need to only supply additional search parameters, without having to send all the query parameters. For example, if the user is searching for a Mercedes-Benz dealership, the request can include only indication of narrowed up categories while the business directory server 112 will store the general categories in which the search will be performed. In yet another embodiment of the invention, the parameters can be cached or stored within the dispatcher module 112, or even within the database 116.

[0038] Information from the context object is transmitted to and from the business directory service 100 using data packets. In a particular embodiment of the invention, the format of the data packets is that of XML packets. Illustrated in FIGS. 5a-c are sample XML definitions, which can be used to transfer responses and requests. Using XML allows for easy storing of context object information and translation of information for displaying in whichever format appropriate for the particular client.

[0039] Three query types can be provided: listing looking, category look up, and category browsing. Query type is specified by the “type” attribute 512 of <request> node 510. The query conditions of the three query types are specified in corresponding node, <listing_lookup> 514, <category_lookup>, and <category_browse> (not shown). The “result_type” attribute 516 allows users to specify how detailed listing results should be. Three results types are defined: Basic, Detail1 and Detail2. The “number_to_return” attribute 518 can be used to limit the maximum number of listing records to be returned.

[0040] A search condition can be combination of the following query criteria: listing name, category name/code and search region. A <listing_name> node can be used to search for listings whose names satisfy requirements such as being equal to, containing, starting with or containing keyword sound like some given keyword. The following example is used to search for listings with name starting with “iron gate”:

```
[0041] <listing_name search_string="iron gate"
search_mode="start_with"/>
```

[0042] A <category> node can be used to search for listings in some specific categories, whose category codes are in the given list or whose names satisfy such requirement as being equal to, containing, starting with, ending with, or sounding like some given string. Either category code or category name can be used to search for categories.

[0043] A <search_region> node can be used to search for listings within some geographic region, which is specified by administration area, such as city, state and postal code, or by distance from some given location.

[0044] In addition to looking up objects, users can browse categories using <category_browse> node. Altogether, there is a wide range of possibilities for types and kinds of searching presented in a general XML schema. The input document schema of a particular embodiment is shown in FIG. 5b.

[0045] Results returned from the database are formatted by the business directory server 112 in a pre-defined output XML schema (FIG. 5c). For a listing lookup query, the business directory server 112 returns a list of matched listing, along with the list of categories that the listing belongs to. For a category lookup query, the business directory server 112 returns a list of matched categories, and for a category browsing query, a listing of the requested category hierarchy is returned.

[0046] In an alternative embodiment of the invention, different XML schemas may be used, as defined by one skilled in the art. In yet another embodiment of the invention, a context object may be passed from the clients to the business directory service 100 in an object state, without converting it to XML definitions. In yet another embodiment of the invention, caching and storing of the information may be used, such that only additional components of the new queries may need to be transferred.

[0047] As discussed above, data from multiple vendors may be used to provide business directory services. The database 114 can contain a wide range of data formats and types. Illustrated in FIGS. 6a-6c are database schema for the business directory service according to one embodiment of the invention. There are three types of tables in the database schema: vendor profile table (FIG. 6a), category table (FIG. 6b), and business listing table (FIG. 6c). Vendor profile table stores the data vendor dependent information. Category tables store the category hierarchy information, while the business listing tables store the actual business directory listing information. There can be multiple category tables and listing tables for multiple data vendors. Typically, each data vendor will have one category table and one listing table to store its own category and business listing information.

[0048] The vendor profile table (FIG. 6a) describes how the actual category and listing data are stored for different vendors. Each row contains information about the category and listing table names for a data vendor and the country to which the data belongs. If a vendor provides data for multiple countries, it can have multiple rows in this table. Thus if INFOUSA is the data supplier for U.S. business directory information, then in the vendor table there will be an entry with the following values: ["INFOUSA", "US", "CATEGORY_INFOUSA", "LISTING_INFOUSA"].

[0049] The listing table stores information of the business listings. Table structure for different data suppliers can be

different, and illustrated in FIG. 6b is one way of structuring table columns. The category table (FIG. 6c) stores category names, SIC codes (or codes for any other nomenclature), and the category hierarchy. In an alternative embodiment of the invention, different database schema can be used. In yet another embodiment of the invention, data from different vendors can be filtered and combined into one coherent set of tables.

[0050] Referring now to FIG. 7, illustrated there is a flow chart illustrating operation of the business directory server module 112. As discussed above, the business directory server module receives client requests (step 710), parses them from the XML format (step 712), finds proper data vendors for the request (step 714), creates an appropriate SQL database query (step 716), which is sent to the database 114 in step 718. After results are returned from the database 114, business directory server module 112 formats the results and returns them to dispatcher module 110 (step 720). The business directory server module can be threaded, such that a separate thread is spawned for each request. In another embodiment of the invention, several business directory server modules can be available at once, listening on the communication ports for receiving the client requests.

[0051] Those of ordinary skill in the art should recognize that methods for the business directory service may be embodied in a computer program product that includes a computer usable medium. For example, such a computer usable medium can include a readable memory device, such as a solid state memory device, a hard drive device, a CD-ROM, a DVD-ROM, or a computer diskette, having stored computer-readable program code segments. The computer readable medium can also include a communications or transmission medium, such as a bus or a communications link, either optical, wired, or wireless, carrying program code segments as digital or analog data signals.

[0052] While the system has been particularly shown and described with references to particular embodiments, it will be understood by those of ordinary skill in the art that various changes in form and details may be made without departing from the scope of the invention encompassed by the appended claims. For example, the methods of the invention can be applied to various environments, and are not limited to the described environment.

What is claimed is:

1. A computer-implemented method for providing business directory services from a networked client, the method comprising:

receiving a search parameter from a user;

repeating, until a satisfactory level of granularity is reached:

sending the search parameter to a business directory server;

receiving, from the business directory server, results of the search responsive to the search parameter; and

receiving, from the user, an additional search parameter for searching within the search results.

2. The method of claim 2, further comprising displaying the search results to the user in such a way as to simplify selection of the additional search parameter.

3. The method of claim 2, wherein the client is at least one of: an email client, a web-based client, and a wireless device client.

4. The method of claim 3, wherein the wireless device client is a cellular phone.

5. The method of claim 1, wherein the search parameter is based at least in part on user preferences stored in the client.

6. The method of claim 5, wherein the user preferences comprise preferences for initial geographic location around which to base the search.

7. The method of claim 6, wherein the initial geographic location is determined based on a location of the client.

8. The method of claim 1, wherein the search parameter comprises a search type parameter, and wherein the search type parameter can be at least one of the following: find a listing, browse listings, and browse a category.

9. The method of claim 8, wherein the search parameter comprises searching within a selected distance from a selected geographical object.

10. The method of claim 1, wherein the search parameter is stored in a context object.

11. The method of claim 10, wherein the context object is encoded in an XML schema.

12. A networked client for providing business directory services, the client comprising:

a user interface for receiving a search parameter from a user;

an encoding module for encoding the search parameter in a context object;

a communications module for transferring the search parameter to a business directory server;

a user interface for displaying search results received from the business directory server; and

a user interface for receiving a second search parameter from a user for a search within the search results received from the business directory server.

13. The networked client of claim 12, wherein the search parameter comprises a search type parameter, and wherein the search type parameter can be at least one of the following: find a listing, browse listings, and browse a category.

14. The networked client of claim 13, wherein the search parameter further comprises a combination of search types.

15. The networked client of claim 12, wherein the client is at least one of the following: an email client, a web-based client, and a wireless device client.

16. A computer program product for providing business directory services to a user, comprising:

a computer readable medium;

instructions recorded on the computer readable medium, including instructions for:

receiving a search parameter from a user; and

repeating, until a satisfactory level of granularity is reached:

sending the search parameter to a business directory server,

receiving, from the business directory server, results of the search responsive to the search parameter, and

receiving, from the user, an additional search parameter for searching within the search results.

17. The computer program product of claim 16, wherein the instructions recorded on the computer readable medium further comprise instructions for displaying the search results to the user in such a way as to simplify selection of the additional search parameter.

18. The computer program product of claim 16, wherein the computer-readable medium is enclosed in a wireless communications device.

19. The computer program product of claim 16, wherein the search parameter is based at least in part on user preferences stored in the client.

20. The computer program product of claim 19, wherein the user preferences comprise preferences for initial geographic location around which to base the search.

21. A networked client for providing business directory services, the client comprising:

means of receiving a search parameter from a user; and

means for repeating, until a satisfactory level of granularity is reached, operations by:

means for sending the search parameter to a business directory server,

means for receiving, from the business directory server, results of the search responsive to the search parameter, and

means for receiving, from the user, an additional search parameter for searching within the search results.

22. The networked client of claim 21, wherein the search parameter comprises a type parameter, and wherein the type parameter is at least one of the following: lookup a listing, browse listings, and browse a category.

23. The networked client of claim 21, wherein the client is embodied in a wireless device.

24. A computer-implemented method for providing business directory services, the method comprising:

receiving, by a dispatcher module, a search request containing a context object;

transferring the search request from the dispatcher module to a business directory server module;

searching a database responsive to at least one search parameter encoded in the context object; and

encoding results of the database search in a second context object.

25. The computer-implemented method of claim 24, wherein the context object is encoded in an XML schema.

26. The computer-implemented method of claim 25, wherein the second context object is encoded in the XML schema.

27. The computer-implemented method of claim 25, wherein the database contains data from at least two vendors.

28. The computer-implemented method of claim 27, further comprising selecting an appropriate data vendor based on the search parameter.

29. The computer-implemented method of claim 27, wherein the data from the at least two vendors are integrated into one database schema.

30. A computer-implemented business directory server, the business directory server comprising:

a dispatcher module adapted to receive search parameters within a context object from a client;

a business directory server module for processing search requests containing context objects forwarded to it by the dispatcher module; and

a database containing business directory data.

31. The business directory server of claim 30, wherein the search parameters comprise a type parameter, and wherein the type parameter is at least one of the following: lookup a listing, browse listings, and browse a category.

32. The business directory server of claim 30, wherein the database contains data from at least two vendors.

33. The business directory server of claim 30, wherein the context object is encoded in a request using an XML schema.

34. The business directory server of claim 30, wherein the business directory server module is further adapted to convert results of the search into a second context object.

35. The business directory server of claim 34, wherein the second context object is encoded in an XML schema before transfer to the client.

36. A system for providing business directory services, the system comprising:

means for receiving, by a dispatcher module, a search request containing a context object;

means for transferring the search request from the dispatcher module to a business directory server module;

means for searching a database responsive to at least one search parameter encoded in the context object; and

means for encoding results of the database search in a second context object.

37. The system of claim 36, wherein the means for encoding results of the database search in a second context object further comprise means for encoding the second context object in an XML schema.

38. A computer program product for providing business directory services, comprising:

a computer readable medium;

instructions recorded on the computer readable medium, including instructions for:

receiving, by a dispatcher module, a search request containing a context object;

transferring the search request from the dispatcher module to a business directory server module;

searching a database responsive to at least one search parameter encoded in the context object; and

encoding results of the database search in a second context object.

39. The computer program product of claim 38, wherein the instructions recorded on the computer readable medium further comprise instructions for encoding the second context object in an XML schema.

* * * * *